

Network Working Group  
Request for Comments: 5234  
STD: 68  
Obsoletes: 4234  
Category: Standards Track

D. Crocker, Ed.  
Brandenburg InternetWorking  
P. Overell  
THUS plc.  
January 2008

## Расширенная спецификация синтаксиса Бэкуса-Наура (ABNF)

### Augmented BNF for Syntax Specifications: ABNF

#### Статус документа

В этом документе содержится спецификация протокола, предложенного сообществу Internet. Документ служит приглашением к дискуссии в целях развития и совершенствования протокола. Текущее состояние стандартизации протокола вы можете узнать из документа Internet Official Protocol Standards (STD 1). Документ может распространяться без ограничений.

#### Аннотация

Технические спецификации Internet зачастую требуют использования формального синтаксиса. За долгие годы модифицированная версия формы Бэкуса-Наура (BNF<sup>1</sup>), названная ABNF<sup>2</sup>, приобрела популярность во множестве спецификаций Internet. В данном документе содержится спецификация ABNF. Эта форма сочетает компактность и простоту с достаточно мощными средствами представления. Различия между стандартной формой BNF и ABNF включают правила именования, повторения, варианты, независимость от порядка (order-independence) и диапазоны значений. Данная спецификация также включает дополнительные определения правил и кодирования для основы лексического анализатора типов, используемого в нескольких спецификациях Internet.

## Оглавление

1. Введение.....	1
2. Определения правил.....	2
2.1 Именованые правил.....	2
2.2 Форма правил.....	2
2.3 Терминальные значения.....	2
2.4 Внешнее кодирование.....	3
3. Операторы.....	3
3.1 Конкатенация: Rule1 Rule2.....	3
3.2 Варианты: Rule1 / Rule2.....	3
3.3 Дополнительные варианты: Rule1 =/ Rule2.....	3
3.4 Диапазоны вариантов: %c## ##.....	3
3.5 Упорядоченная группа: (Rule1 Rule2).....	3
3.6 Переменное число повторов: *Rule.....	4
3.7 Заданное число повторов: nRule.....	4
3.8 Необязательная последовательность: [RULE].....	4
3.9 Комментарий: ; Comment.....	4
3.10 Старшинство операторов.....	4
4. ABNF-определение для ABNF.....	4
5. Вопросы безопасности.....	5
6. Литература.....	5
6.1 Нормативные документы.....	5
6.2 Дополнительная литература.....	5
Приложение А. Благодарности.....	5
Приложение В. Основы ABNF для ABNF.....	5
В.1 Базовые правила.....	5
В.2 Общие правила кодирования.....	6

## 1. Введение

В технических спецификациях Internet часто требуется определять формальный синтаксис и можно выбирать ту или иную нотацию, которую авторы считают полезной. За долгие годы модифицированная версия формы Бэкуса-Наура (BNF), названная ABNF, приобрела популярность во множестве спецификаций Internet. Эта форма сочетает компактность и простоту с достаточно мощными средствами представления. В раннюю эпоху Agranet каждая спецификация использовала свое определение ABNF. К таким определениям относится спецификация электронной почты [RFC733] и более поздний документ [RFC822], которые стали основой для последующих определений ABNF. Текущий документ разделяет эти определения для того, чтобы можно было независимо ссылаться на них. Кроме того в этом документе внесены некоторые изменения и усовершенствования.

Различия между стандартной формой BNF и ABNF включают правила именования, повторения, варианты, независимость от порядка (order-independence) и диапазоны значений. В Приложении В даются определения правил и

<sup>1</sup>Backus-Naur Form.

<sup>2</sup>Augmented BNF - расширенная форма Бэкуса-Наура. Прим. перев.

кодирования для основы лексического анализатора типа, используемого в нескольких спецификациях Internet. Они приводятся для удобства и отделения от метаязыка, определенного в основной части данного документа, и его формального статуса.

## 2. Определения правил

### 2.1 Именованние правил

Имя правила является лишь именем как таковым, т. е., последовательностью символов, начинающейся с буквы<sup>1</sup> или цифры, за которой следует комбинация букв, цифр и знаков дефиса (-).

Примечание: строчные и прописные буквы в именах правил не различаются.

Имена `<rulename>`, `<RuleName>`, `<RULENAME>` `<rUIENamE>` указывают на одно правило.

В отличие от исходной формы BNF угловые скобки (< и >) не являются обязательными. Однако в такие скобки могут заключаться имена правил, когда нужно подчеркнуть использование имени правила. Это обычно относится к именам правил, указанным в свободной форме, или для выделения имен правил включенных в строку без разделения символами пробела, как показано ниже при обсуждении повторов.

### 2.2. Форма правил

Правила определяются следующим способом:

```
name = elements crlf
```

где `<name>` - имя правила, `<elements>` - одно или множество имен правила или терминальных спецификаций, а `<crlf>` - индикатор завершения строки (возврат каретки с последующим переводом строки). Знак равенства отделяет имя от определения правила. Элементы формируют последовательность из одного или множества имен правил и/или определений значений, объединенных с помощью различных операторов, определяемых в данном документе, таких, как повторы или варианты.

Для наглядности определения правил выровнены по левому краю. Если правило не помещается на одной строке, используются знаки продолжения строки. Выравнивание по левому краю и сдвиг от края определяются первой строкой правила ABNF, чтобы правила можно было легко отличить от остального текста документа.

### 2.3. Терминальные значения

Правила преобразуются в строки терминальных значений, иногда называемых символами. В ABNF символ - это просто неотрицательное целое число. В некоторых случаях задается специфическое отображение (кодирование) значений в набор символов (такой, как ASCII).

Терминалы задаются одной или множеством цифр с явно заданным основанием. В настоящее время определены следующие варианты оснований чисел:

```
b          = binary (двоичное)
d          = decimal (десятичное)
x          = hexadecimal (шестнадцатеричное)
```

Следовательно, записи:

```
CR          = %d13
CR          = %x0D
```

задают, соответственно, десятичное и шестнадцатеричное представление символа возврата каретки [US-ASCII].

Связанные (concatenated) строки таких значений задаются в компактной форме с использованием символа точки (.) в качестве разделителя. Следовательно,

```
CRLF       = %d13.10
```

ABNF разрешает указывать текстовые строки непосредственно, помещая текст в двойные кавычки.

```
command    = "command string"
```

Текстовые строки интерпретируются как связанное множество печатных символов.

Примечание: регистр букв текстовых строках ABNF не принимается во внимание, а в качестве набора символов используется US-ASCII.

Следовательно,

```
rulename   = "abc"
```

и

```
rulename   = "aBc"
```

будут соответствовать «abc», «Abc», «aBc», «abC», «ABc», «aBC», «AbC» и «ABC».

Для задания правила, в котором регистр символов принимается во внимание, символы следует задавать по отдельности.

Например,

```
rulename   = %d97 %d98 %d99
```

или

```
rulename   = %d97.98.99
```

будут соответствовать только строкам, содержащим лишь строчные буквы abc.

<sup>1</sup>Здесь и далее в тексте документа термин "буква" означает символы латинского алфавита. *Прим. перев.*

## 2.4. Внешнее кодирование

Внешнее представление символов терминальных значений будет меняться в соответствии с условиями среды хранения и передачи. Следовательно, одна и та же грамматическая конструкция ABNF может иметь различное внешнее кодирование (например, одно представление для 7-битовой среды US-ASCII, другое для двоичной среды на базе октетов, третье для 16-битовой среды Unicode). Детали кодирования выходят за пределы ABNF, хотя в Приложении В (Основы) приводятся определения для 7-битовой среды US-ASCII, как наиболее распространенной в Internet.

Разделение внешнего кодирования и синтаксиса предназначено для того, чтобы могли использоваться дополнительные среды кодирования для одного и того же синтаксиса.

## 3. Операторы

### 3.1. Конкатенация: Rule1 Rule2

Правило может определять простую, упорядоченную строку значений (например, слияние последовательных символов) путем перечисления последовательности имен правил. Например,

```
foo      = %x61          ; a
bar      = %x62          ; b
mumble   = foo bar foo
```

В результате правило <mumble> будет соответствовать строке строчных букв "aba".

**Пробельные символы:** Конкатенация является основой модели разбора ABNF. Строка непрерывных символов (значений) разбирается согласно правилам, определенным в ABNF. Для спецификаций Internet в силу исторических причин допускается свободное «рассеяние» символов пробела и горизонтальной табуляции вокруг основных конструкций (таких, как специальные разделители или строки-атомы).

**Примечание:**

В данной спецификации ABNF не предполагается неявно такого «рассеяния» пробельных символов.

Любая грамматическая конструкция, которая разрешает «рассеяние» пробельных символов вокруг разделителей и сегментов строк, должна указывать это явно. Зачастую бывает полезно обеспечить пустое пространство в основных правилах, которые используются среди правил вышележащих уровней. «Основные» правила могут формироваться в лексический анализатор или просто быть частью главного набора правил.

### 3.2. Варианты: Rule1 / Rule2

Элементы, разделенные символом дробной черты (/), задают варианты. Следовательно,

```
foo / bar
```

будет принимать значение <foo> или <bar>.

**Примечание:**

Заключенные в двойные кавычки строки букв представляют собой специальную форму задания дополнительных символов и интерпретируются как нетерминальное представление набора комбинаторных строк с содержащимися в них символами в указанном порядке, но с любыми комбинациями строчных и прописных букв.

### 3.3. Дополнительные варианты: Rule1 /= Rule2

Иногда удобно задавать список вариантов в виде фрагментов, т. е. начальному правилу соответствует один или множество вариантов, а последующие определения правил добавляют набор вариантов. Это особенно полезно для создания иной независимой спецификации, которая является производной от того же родительского набора правил, как это часто бывает со списками параметров. ABNF поддерживает такие дополнения с помощью конструкции:

```
oldrule   /= дополнительные-варианты
```

Таким образом набор, правил

```
ruleset   = alt1 / alt2
ruleset   /= alt3
ruleset   /= alt4 / alt5
```

представляет собой то же самое, что

```
ruleset   = alt1 / alt2 / alt3 / alt4 / alt5
```

### 3.4. Диапазоны вариантов: %c##-##

Диапазон вариантов цифровых значений может быть представлен в компактной форме с использованием символа дефиса (-) для индикации диапазона вариантов. Следовательно,

```
DIGIT     = %x30-39
```

является эквивалентом:

```
DIGIT     = "0" / "1" / "2" / "3" / "4" / "5" / "6" / "7" / "8" / "9"
```

Конкатенация цифровых значений и диапазон цифровых значений не могут быть заданы в одной строке. Цифровое значение может использовать нотацию с точками для конкатенации или нотацию с дефисом для задания одного диапазона значений. Следовательно, для задания одного печатного символа между последовательностями завершения строк, можно указать:

```
char-line = %x0D.0A %x20-7E %x0D.0A
```

### 3.5. Упорядоченная группа: (Rule1 Rule2)

Элементы, заключенные в круглые скобки, трактуются как один элемент со **строгим упорядочением**. Таким образом,

```
elem (foo / bar) blat
```

соответствует (elem foo blat) или (elem bar blat), а

```
elem foo / bar blat
```

соответствует (elem foo) или (bar blat).

#### Примечание:

Настоятельно рекомендуется использовать группы взамен вариантов, состоящих из множества имен правил или литералов.

Следовательно, рекомендуется использовать приведенную ниже форму:

```
(elem foo) / (bar blat)
```

Это позволит предотвратить ошибочную интерпретацию при невнимательном чтении.

Упорядоченные группы используются также внутри свободного текста для выделения порядка слов в тексте.

### 3.6. Переменное число повторов: \*Rule

Оператор \*, предшествующий элементу, указывает на повторение. Полная форма имеет вид:

```
<a>*<b>element
```

где <a> и <b> - необязательные десятичные значения, показывающие минимальное (<a>) и максимальное (<b>) число повторов элемента.

По умолчанию для верхнего и нижнего порога используются значения 0 и «бесконечность», поэтому \*<element> позволяет любое число элементов (включая 0), 1\*<element> требует по крайней мере один элемент, 3\*3<element> разрешает в точности три повтора, а 1\*2<element> разрешает 1 или 2 повтора.

### 3.7. Заданное число повторов: nRule

Правило вида:

```
<n>element
```

эквивалентно правилу

```
<n>*<n>element
```

Т. е., допускается в точности <n> включений <element>. Таким образом 2DIGIT представляет собой двухзначное число, а 3ALPHA - строку из трех букв.

### 3.8. Необязательная последовательность: [RULE]

В квадратных скобках указывается необязательная последовательность элементов.

```
[foo bar]
```

эквивалентно

```
*1(foo bar).
```

### 3.9. Комментарий: ; Comment

Точка с запятой (;) служит началом комментария, который продолжается до конца строки. Это обеспечивает простой способ включения в спецификацию полезных примечаний.

### 3.10. Старшинство операторов

Описанные выше операторы имеют разный уровень старшинства (порядок применения). Далее операторы перечислены в соответствии со старшинством - сначала указаны операторы с высшим приоритетом, а последним указан оператор с низшим уровнем старшинства:

Имена правил, текстовые строки, терминальные значения (Rule name, prose-val, Terminal value)

Комментарий (Comment)

Диапазон значений (Value range)

Повтор (Repetition)

Группировка, необязательные последовательности (Grouping, Optional)

Конкатенация (Concatenation)

Варианты (Alternative)

Использование вариантов, произвольно перемешанных с операторами конкатенации, может привести к путанице.

Снова рекомендуется использовать оператор группировки для явного указания сливаемых воедино групп.

## 4. ABNF-определение для ABNF

#### Примечания:

1. Этот синтаксис требует сравнительно строгого форматирования правил. Следовательно, включенный в спецификацию вариант набора правил может потребовать предварительной обработки для того, чтобы он мог быть обработан анализатором ABNF.
2. Синтаксис использует правила, приведенные в Приложении В.

```
rulelist = 1*( rule / (*c-wsp c-nl) )
rule     = rulename defined-as elements c-nl
```

```

; продолжение, если следующая строка начинается с пробела
rulename      = ALPHA *(ALPHA / DIGIT / "-")
defined-as    = *c-wsp ("=" / "=") *c-wsp
; определение базовых правил и дополнительных вариантов
elements      = alternation *c-wsp
c-wsp         = WSP / (c-nl WSP)
c-nl          = comment / CRLF
; комментарий или новая строка
comment       = ";" *(WSP / VCHAR) CRLF
alternation   = concatenation *( *c-wsp "/" *c-wsp concatenation)
concatenation = repetition *(1*c-wsp repetition)
repetition    = [repeat] element
repeat        = 1*DIGIT / (*DIGIT "*" *DIGIT)
element       = rulename / group / option / char-val / num-val / prose-val
group         = "(" *c-wsp alternation *c-wsp ")"
option        = "[" *c-wsp alternation *c-wsp "]"
char-val      = DQUOTE *(%x20-21 / %x23-7E) DQUOTE
; заключенная в кавычки строка SP и VCHAR без DQUOTE
num-val       = "%" (bin-val / dec-val / hex-val)
bin-val       = "b" 1*BIT [ 1*("." 1*BIT) / ("-" 1*BIT) ]
; последовательность объединенных битовых
; значений или один диапазон ONEOF
dec-val       = "d" 1*DIGIT [ 1*("." 1*DIGIT) / ("-" 1*DIGIT) ]
hex-val       = "x" 1*HEXDIG [ 1*("." 1*HEXDIG) / ("-" 1*HEXDIG) ]
prose-val     = "<" *(%x20-3D / %x3F-7E) ">"
; заключенная в угловые скобки последовательность SP и VCHAR
; за исключением правых угловых скобок будет использоваться

```

## 5. Вопросы безопасности

Искренне надеемся, что вопросы безопасности не имеют отношения к этому документу.

## 6. Литература

### 6.1. Нормативные документы

[US-ASCII] American National Standards Institute, "Coded Character Set -- 7-bit American Standard Code for Information Interchange", ANSI X3.4, 1986.

### 6.2. Дополнительная литература

[RFC733] Crocker, D., Vittal, J., Pogram, K., and D. Henderson, "Standard for the format of ARPA network text messages", RFC 733, November 1977.

[RFC822] Crocker, D., "Standard for the format of ARPA Internet text messages", STD 11, RFC 822<sup>1</sup>, August 1982.

## Приложение А. Благодарности

Изначальная спецификация синтаксиса ABNF была приведена в RFC 733. Ken L. Harrenstien из SRI International выполнил работу по кодированию для расширенного BNF, позволившему сделать представление более компактным и ясным.

Этот недавний проект начался как простая работа по выделению части документа RFC 822, который неоднократно использовался в спецификациях, не связанных с электронной почтой, и являлся описанием расширенного варианта BNF. Вместо простого копирования имеющегося текста в отдельный документ, рабочая группа предпочла внимательное рассмотрение преимуществ и недостатков существующей спецификации и других спецификаций, выпущенных за последние 15 лет. Это сделало проект более амбициозным, нежели предполагалось в начале. Интересно отметить, что полученный результат не слишком значительно отличается от оригинала, хотя некоторые решения (типа удаления списков - list notation) стали сюрпризом.

Эта «отдельная» версия спецификации была частью работы группы DRUMS, значительный вклад в которую внесли Jerome Abela, Harald Alvestrand, Robert Elz, Roger Fajman, Aviva Garrett, Tom Harsch, Dan Kohn, Bill McQuillan, Keith Moore, Chris Newman, Pete Resnick и Henning Schulzrinne.

Julian Reschke заслуживает отдельной благодарности за преобразование черного варианта в формат XML.

## Приложение В. Основы ABNF для ABNF

Это приложение включено как удобная основа для грамматических конструкций. Приведенные здесь определения можно использовать как базовый набор правил.

### В.1. Базовые правила

Для некоторых базовых правил (таких, как SP, HTAB, CRLF, DIGIT, ALPHA и т. п.) используются заглавные буквы.

```

ALPHA        = %x41-5A / %x61-7A ; A-Z / a-z
BIT          = "0" / "1"
CHAR         = %x01-7F
; любой 7-битовый символ US-ASCII, за исключением NUL
CR           = %x0D
; возврат каретки
CRLF         = CR LF
; стандартная в Internet последовательность для новой строки

```

<sup>1</sup>Этот документ признан устаревшим и заменен [RFC 2822](#), а тот, в свою очередь, - [RFC 5322](#). Прим. перев.

CTL	=	%x00-1F / %x7F	
			; коды управления
DIGIT	=	%x30-39	
			; цифры 0-9
DQUOTE	=	%x22	
			; " (двойные кавычки)
HEXDIG	=	DIGIT / "A" / "B" / "C" / "D" / "E" / "F"	
HTAB	=	%x09	
			; символ горизонтальной табуляции
LF	=	%x0A	
			; перевод строки
LWSP	=	*(WSP / CRLF WSP)	
			; Использование этого правила позволяет включать строки,
			; содержащие только символы пробела, которые больше не
			; считаются допустимыми в заголовках электронной почты и
			; могут вызывать проблемы интероперабельности в ином контексте
			; Не используйте это правило при определении почтовых
			; заголовков и с осторожностью применяйте в другом контексте
OCTET	=	%x00-FF	
			; 8 битов данных
SP	=	%x20	
VCHAR	=	%x21-7E	
			; видимые (печатные) символы
WSP	=	SP / HTAB	
			; пробельные символы

## В.2. Общие правила кодирования

Для внешнего представления данных используется 7-битовая кодировка US-ASCII с нулевым значением старшего бита. Строки значений используют «сетевой порядок байтов», при котором старший (наиболее значимый байт) указывается слева и передается через сеть первым.

### Адреса авторов

**Dave Crocker (редактор)**  
Brandenburg InternetWorking  
675 Spruce Dr.  
Sunnyvale, CA 94086  
US  
Phone: +1.408.246.8253  
EMail: [dcrocker@bbiw.net](mailto:dcrocker@bbiw.net)

**Paul Overell**  
THUS plc.  
1/2 Berkeley Square,  
99 Berkeley Street  
Glasgow G3 7HR  
UK  
EMail: [paul.overell@thus.net](mailto:paul.overell@thus.net)

### Перевод на русский язык

**Николай Малых**

[nmalykh@protokols.ru](mailto:nmalykh@protokols.ru)

### Полное заявление авторских прав

#### Copyright (C) The IETF Trust (2008).

К этому документу применимы права, лицензии и ограничения, указанные в BCP 78, и, за исключением указанного там, авторы сохраняют свои права.

Этот документ и содержащаяся в нем информация представлены "как есть" и автор, организация, которую он/она представляет или которая выступает спонсором (если таковой имеется), Internet Society и IETF отказываются от каких-либо гарантий (явных или подразумеваемых), включая (но не ограничиваясь) любые гарантии того, что использование представленной здесь информации не будет нарушать чьих-либо прав, и любые предполагаемые гарантии коммерческого использования или применимости для тех или иных задач.

### Интеллектуальная собственность

IETF не принимает какой-либо позиции в отношении действительности или объема каких-либо прав интеллектуальной собственности (Intellectual Property Rights или IPR) или иных прав, которые, как может быть заявлено, относятся к реализации или использованию описанной в этом документе технологии, или степени, в которой любая лицензия, по которой права могут или не могут быть доступны, не заявляется также применение каких-либо усилий для определения таких прав. Сведения о процедурах IETF в отношении прав в документах RFC можно найти в BCP 78 и BCP 79.

Копии раскрытия IPR, предоставленные секретариату IETF, и любые гарантии доступности лицензий, а также результаты попыток получить общую лицензию или право на использование таких прав собственности разработчиками или пользователями этой спецификации, можно получить из сетевого репозитория IETF IPR по ссылке <http://www.ietf.org/ipr>.

IETF предлагает любой заинтересованной стороне обратить внимание на авторские права, патенты или использование патентов, а также иные права собственности, которые могут потребоваться для реализации этого стандарта. Информацию следует направлять в IETF по адресу [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).